BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B69 03097

SUBJECT: Subroutines HUNT3 and HUNT4 for Trajectory Targeting with BCMASP - Case 610

DATE: March 26, 1969

FROM: P. H. Whipple

ABSTRACT

Subroutines HUNT3 and HUNT4 have been developed to enable the Bellcomm Apollo Simulation Program to simulate launch trajectories that use yaw steering during the ascent to orbit. HUNT3 determines the values of three trajectory shaping parameters such that constraints on three orbital elements are satisfied. Similarly, HUNT4 determines values for four trajectory shaping parameters to satisfy constraints on four orbital elements at insertion.

One application of HUNT3 is the determining of values for two pitch parameters and one yaw parameter that will result in an orbit with a specified altitude, flight path angle, and orbit inclination at insertion. If HUNT4 is used, the position of the line of nodes of the achieved orbit can also be specified by introducing a second yaw parameter.

Although these subroutines have been developed for simulating Saturn IB launches into earth orbit, they could be adapted to simulations of missions that use other launch vehicles and to other trajectory targeting problems.

(NASA-CR-103984) SUBROUTINES HUNT3 AND HUNT4 FOR TRAJECTORY TARGETING WITH BCMASP (Bellcomm, Inc.) 20 p

N79-71880

Unclas
00/13 11519

FF No. 602

CR-103984 (PAGES) (CODE)
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)

[redacted]

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B69 03097

SUBJECT: Subroutine HUNT3 and HUNT4 for Trajectory Targeting with BCMASP - Case 610

DATE: March 26, 1969

FROM: P. H. Whipple

MEMORANDUM FOR FILE

I. INTRODUCTION

The simulation of launch trajectories with the Bellcomm Apollo Simulation Program (BCMASP) requires the determination of values of trajectory shaping parameters that will result in achieving a specified orbit. This has usually been done by considering one (or two) pitch-plane shaping parameters at a time and determining values for these parameters such that one (or two) trajectory constraints are satisfied. Subroutines HUNT1 and HUNT2 have been used for these purposes. For Apollo Applications Program Mission analysis, it has been necessary to extend these procedures to permit the determination of yaw parameters in addition to the pitch parameters. Subroutines HUNT3 and HUNT4 have been developed for this purpose.

This memorandum describes these subroutines and gives examples of how they are used.

II. DISCUSSION

A. Simulation of Launch into an Orbit of Specified Inclination

Launching at the optimum launch azimuth gives the maximum payload into orbit and results in a launch trajectory that requires essentially no yaw steering of the launch vehicle. For high inclination orbits, the optimum launch azimuth is often unacceptable because of range safety limitations. It is necessary to launch at an azimuth within the range safety limits and initiate a yaw maneuver during the ascent to achieve the desired inclination. For a Saturn IB launch simulation, the launch vehicle rises vertically from the launch pad for the first ten seconds of flight. An initial pitch or kick angle is then instantaneously applied and a gravity turn trajectory is followed until approximately ten seconds before shutdown of the first stage. An inertial attitude hold is then maintained during the last ten seconds of the first stage burn, through the coast period between first stage shutdown and second stage ignition, and for approximately the first 30 seconds of the second stage burn. To insure that the simulation is a reasonable approximation to a realistic launch trajectory, an altitude constraint at this point is enforced by HUNT1, using the kick angle as the trajectory shaping parameter.

At the end of the inertial hold period, an instantaneous increment in pitch angle is assumed and yaw and pitch programs are initiated and continued until second stage shutdown. Constraints on radius, flight path angle, and orbit inclination at second stage shutdown are enforced by HUNT3, using as trajectory shaping parameters the pitch and yaw rates and the increment in pitch angle.

HUNT3 may also be used to achieve a specified orbit inclination when yaw steering of the launch vehicle is not done. In this case, the launch azimuth can be used as a trajectory shaping parameter.

B. Rendezvous Mission Launch Simulation

The spacecraft launched second in a rendezvous mission must be accurately inserted into a specific orbit to insure adequate phasing capability and a minimum plane change requirement. At orbital insertion, the spacecraft must not only have a specific altitude, flight path angle, and orbit inclination, but the position of the orbital plane must be accurately controlled. This is usually done in BCMASP by specifying the longitude of the descending node of the orbit, measured eastward from the inertial longitude of the launch pad at launch.

The sequence of events for the launch into orbit is the same as in the previous example with one exception. At the end of the inertial attitude hold period about 30 seconds after second stage ignition, an increment in yaw angle is assumed in addition to the increment in pitch angle. These together with the pitch and yaw rates give four trajectory shaping parameters for HUNT4 to use in enforcing the four trajectory constraints at orbital insertion. It is sometimes desirable to use launch azimuth in place of the increment in yaw angle.

C. Determination of Values of the Trajectory Shaping Parameter

In trajectory targeting with BCMASP, the HUNT subroutines are used to determine values of the trajectory shaping parameters that will produce a launch trajectory satisfying the trajectory constraints. The basic mathematical approach used in HUNT3 and HUNT4 is given in Reference 1. A brief summary is given below.

In general the n-dimensional HUNT problem is to find the values of n independent variables so that n dependent variables will have specified values within specified tolerances. In applying this to the development of a trajectory, the independent variables are the trajectory shaping parameters and the dependent variables are the trajectory elements upon which constraints are imposed. The mathematical statement of the problem is

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, \ldots, x_n) \\
y_2 &= f_2(x_1, x_2, \ldots, x_n) \\
&\vdots \\
y_n &= f_n(x_1, x_2, \ldots, x_n)
\end{aligned}
$$

where the x's are the independent variables and the y's are dependent variables. The functions are not usually known in analytic form and numerical integration is required to evaluate the y's.

If the y's are single-valued functions of the x's, then the problem can be restated as

$$
\begin{aligned}
x_1 &= g_1(y_1, y_2, \ldots, y_n) \\
&\vdots \\
x_n &= g_n(y_1, y_2, \ldots, y_n)
\end{aligned}
$$

and the total differential for x_k can be written

$$dx_k = \sum_{i=1}^{n} \frac{\partial x_k}{\partial y_i} dy_i \quad .$$

If the functions are assumed linear, then

$$\Delta x_k = \sum_{i=1}^{n} \frac{\partial x_k}{\partial y_i} \Delta y_i \quad . \tag{1}$$

This is a relationship between small changes in all of the dependent variables with a small change in one independent variable.

In generating a trajectory, a set of shaping parameters (x's) is assumed and the trajectory is integrated over the region of interest. The resulting values for the trajectory elements (y's) are then compared to the trajectory constraints. If the error in each element is less than a specified tolerance, no further work is required by the HUNT subroutine. If the error is unacceptably large for one or more of the elements, another set of shaping parameters is determined and the trajectory is integrated again. A new set of shaping parameters is obtained by incrementing the set previously used with the increments obtained from Equation (1). The increments of the y variables of (1) are taken to be the errors in the trajectory elements. This assumes that the coefficients or partial derivatives have previously been determined.

If the relationships between the dependent and independent variables were perfectly linear, only one correction to the shaping parameters would be required to enforce the trajectory constraints. However, due to inherent non-linearities, several iterations are almost always required before all of the trajectory elements achieve their required values. The non-linearities also necessitate updating the partial derivatives of Equation (1) after each iteration.

The derivation of the expressions for the partial derivatives is in Reference 1 and only a summary will be given here. For an n-dimensional HUNT problem, n+1 trajectory integrations are required, each with a different set of shaping parameter values. This gives n sets of Δx and Δy for each x and y. The partial derivatives are computed from the following expression:

$$[A] = \left\{[\Delta y]^{-1}[\Delta x]\right\}^T$$

where

$[A]$ is the matrix of partial derivatives,
$[\Delta x]$ is the matrix of independent variable increments,
$[\Delta y]$ is the matrix of dependent variable increments.

$$a_{ij} = \frac{\partial x_i}{\partial y_j}$$

$$\Delta x_{kl} = x_l(\text{k+1st integration}) - x_l(\text{kth integration})$$

$$\Delta y_{kl} = y_l(\text{k+1st integration}) - y_l(\text{kth integration}).$$

The latest n+1 trajectory integrations are always used in computing the partial derivatives.

In applying HUNT3 and HUNT4, additional problems were encountered because of the non-linearities. In selecting the initial estimates for the shaping parameters, it is often very difficult to choose a set of accurate values. Therefore, the HUNT subroutines must be able to overcome large initial errors. However, the eventual convergence of the shaping parameters to a set of values that will result in the required orbit is dependent upon the initial set of values assigned to the shaping parameters. If the initial estimates are too much in error, the shaping parameters will not converge to reasonable values but instead will diverge to meaningless values giving no useful result. This problem has been overcome with HUNT4 by initially converting the four-dimensional problem into two two-dimensional problems. The two-dimensional HUNT problem is much less sensitive to the accuracy of the initial estimates in the shaping parameters. By solving for approximate values of the shaping parameters two at a time, a much better initial estimate for the HUNT4 logic is obtained and convergence of the four-dimensional problem is easily achieved.

In solving the two two-dimensional problems, the pitch parameters are grouped together and the yaw parameters are grouped together. For trajectories where the errors in the trajectory elements are relatively small, the pitch parameters are weakly coupled to the yaw parameters and the solutions to these two two-dimensional problems may be found simultaneously. However, for trajectories where the errors are very large, the coupling is much stronger and one of the two-dimensional problems must be solved before the other. To determine if the more efficient method of simultaneous solutions is feasible, the initial errors are tested when Subroutine HUNT4 is entered. For example, if the initial errors at orbital insertion in radius, flight path angle, orbit inclination, and longitude of the descending node are less than 1,000,000 feet, six degrees, five degrees, and eleven degrees respectively, simultaneous solutions of the two two-dimensional problems are obtained. These numbers are based on the results of several simulations but are subject to modification as more experience is gained. This scheme used in HUNT4 can be summarized as follows:

HUNT4

Is the initial error in radius, flight path angle, orbit inclination or longitude of the descending node very large?

NO

Simultaneously,
enforce radius and flight path angle constraints by varying pitch parameters;
enforce inclination and node longitude constraints by varying yaw parameters.

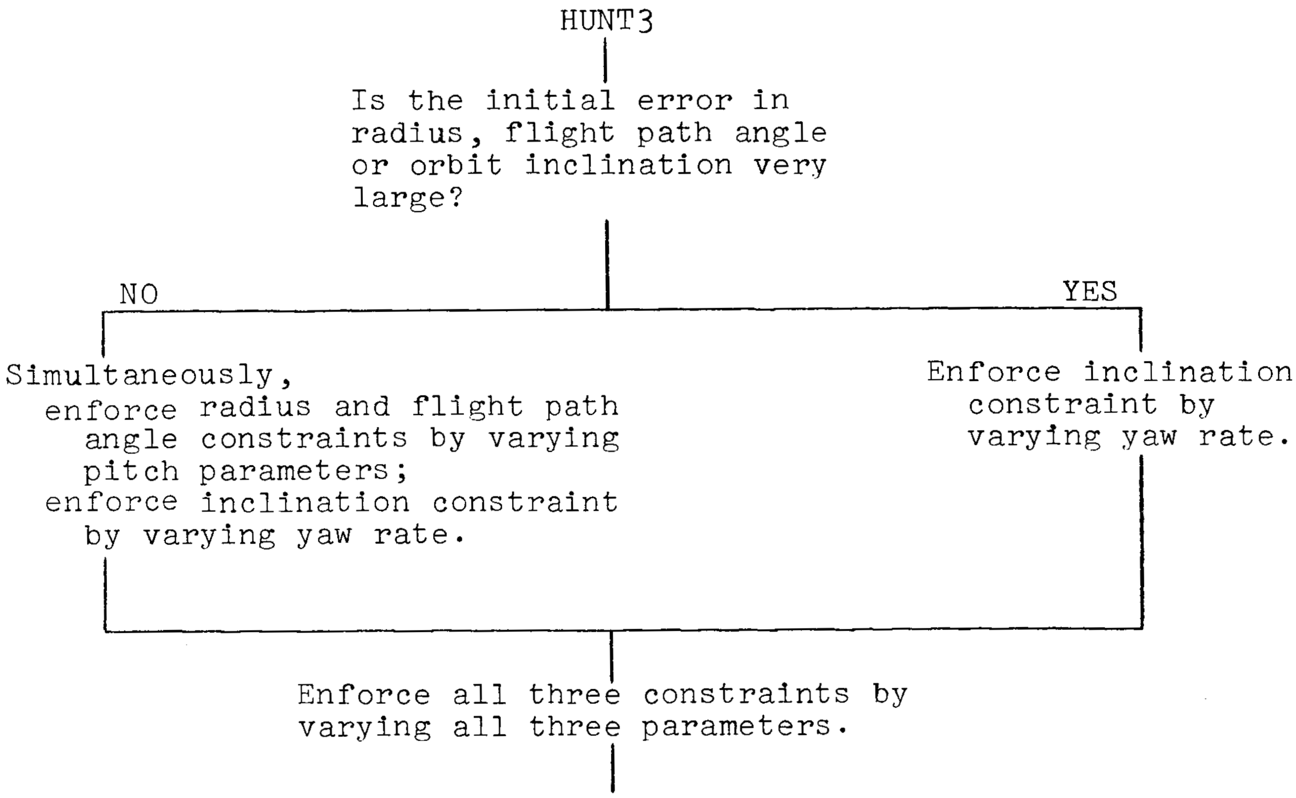
YES

Enforce inclination and node longitude constraints by varying yaw parameters.

Enforce radius and flight path angle constraints by varying pitch parameters.

Enforce all four constraints by varying all four parameters.

A similar scheme was implemented in HUNT3. If the initial errors in radius, flight path angle, and orbit inclination are less than 1,000,000 feet, five degrees, and five degrees respectively, solutions for a two-dimensional problem and a one-dimensional problem are found simultaneously. The two-dimensional problem is the enforcement of the radius and flight path angle constraints by varying the two pitch parameters and the one-dimensional problem is the enforcement of the inclination constraint by varying the yaw rate. After these solutions are found, an improved set of parameter values is available for the HUNT3 logic to use in solving the three-dimensional problem. If any of the initial errors are excessive, the one-dimensional problem must be solved first. Experience has shown that in this case, the solution of the two-dimensional problem is unnecessary before proceding to the three-dimensional problem. The scheme for HUNT3 is as follows:



Listings of HUNT3 and HUNT4 are given in Appendices I and II.

III. SUMMARY

For Apollo Applications Program mission analysis, the simulation of launch trajectories has necessitated extending the targeting capability of BCMASP. Two new subroutines, HUNT3 and HUNT4, have been developed to allow the enforcement of constraints at launch vehicle cutoff on orbit inclination, and orbit inclination and longitude of the descending node respectively, in addition to the constraints on radius and flight path angle. Some of the basic elements of these new subroutines are similar to the simpler subroutines, HUNT1 and HUNT2, which have been used in BCMASP. Convergence problems resulting from non-linearities required the inclusion of additional logic to determine partial solutions of the targeting problem before a complete solution could be found.

While HUNT3 and HUNT4 have been developed for targeting Saturn IB launch trajectories, they could be readily adapted to a variety of three or four-dimensional targeting programs, and to missions using other launch vehicles.

IV. ACKNOWLEDGEMENT

I wish to thank Miss J. C. Gurasich for her assistance in programming HUNT3 and HUNT4.

P. H. Whipple

1025-PHW-dcs

Attachments

BELLCOMM, INC.

REFERENCE

1. Bellcomm Apollo Simulation Program - General Purposes Targeting Programs - HUNT1 and HUNT2, G. Bamesberger, BTL MF6-4264-2, January 3, 1966.

APPENDIX I

```
      SUBROUTINE HUNT3(X,Y,Z,U,V,W,UD,VD,WD,QU,QV,QW,I1,I2,SENS)
C
      DIMENSION SENS(17)
C
C     SUBROUTINE HUNT3 FINDS VALUES OF X AND Y AND Z SUCH THAT U AND V
C     AND W ARE EQUAL TO UD AND VD AND WD WITHIN A TOLERANCE OF QU AND
C     QV AND QW, WHERE U, V, AND W ARE DEPENDENT VARIABLES DEFINED
C     AT EVENT I2 AND X, Y, AND Z ARE CONTROL VARIABLES FOR THE FLIGHT
C     FROM EVENT I1 TO EVENT I2.  IF THE SOLUTION DIVERGES, HUNT3 WILL
C     STOP RUN AFTER 20 ITERATIONS.
C
C
      DATA ZERO/0.0/
C
C     SENS(1) - FLAG TO INDICATE SENSITIVITY IS AVAILABLE
C     SENS(2) - A11   ) - SENSITIVITY MATRIX OR INITIAL STEP DESIRED
C     SENS(3) - A21   )
C     SENS(4) - A31
C     SENS(5) - A12
C     SENS(6) - A22
C     SENS(7) - A32   )
C     SENS(8) - A13
C     SENS(9) - A23
C     SENS(10) - A33   )
C
C INITIALIZE
C
      K=1
      L=0
      M=0
      N=0
      MN=0
      FUMAX=1000000.
      FVMAX= 5.0
      FWMAX= 5.0
      IF(SENS(1).NE.0.) GO TO 2
      DO1 I=2,10
 1    SENS(I)=0.
 2    IF(SENS(15).NE.0.)GO TO 4
      DO 3 I=11,14
 3    SENS(I)=0.
C
 4    IF(SENS(17).NE.0.)GO TO 10
 5    SENS(16)=0.
C
C  INTEGRATE TO EVENT I2 AND EVALUATE ERRORS
 10   CALL FLTINT(I1,I2)
      U=U+ZERO
      UD=UD+ZERO
      V=V+ZERO
      VD=VD+ZERO
      W=W+ZERO
      WD=WD+ZERO
      FU=U-UD
      FV=V-VD
```

```
      FW=W-WD
      IF(MN) 15,16,17
 15   WRITE(6,103) K,X,EU,SENS(16),MN,Y,EV,Z,EW
      GO TO 18
 16   WRITE(6,101)K,X,EU,Y,EV,Z,EW
      GO TO 18
 17   WRITE(6,105) K,X,EU,SENS(2),SENS(5),SENS(8),MN,Y,EV,SENS(3),SENS(6
     .),SENS(9),L,Z,EW,SENS(4),SENS(7),SENS(10)
C
C
 18   IF((ABS(EU).LT.QU).AND.(ABS(EV).LT.QV).AND.(ABS(EW).LT.QW))RETURN
      IF(K.GE.30) GO TO 9
C
C     COMPUTE SENSITIVITIES
      DU3=DU2
      DU2=DU1
      DU1=SU-U
      DV3=DV2
      DV2=DV1
      DV1=SV-V
      DW3=DW2
      DW2=DW1
      DW1=SW-W
      IF(MN) 20,19,50
 19   IF((ABS(EU).LT.EUMAX).AND.(ABS(EV).LT.EVMAX).AND.(ABS(EW).LT.EWMAX
     .))GO TO 48
      MN=-1
C
C
C
C        SOLVE  (Z'W)  PROBLEM  BEFORE SOLVING  (X,Y,Z'U,V,W)  PROBLEM
C
 20   IF(ABS(EW).LT.QW   )GO TO 29
      IF(K.GE.2)GO TO 23
      IF(SENS(17).NE.0.)GO TO 24
C
 21   DX1=0.0
      DY1=0.0
      DZ1=-.01*Z
      GO TO 8
C
 23   IF(DW1.EQ.0.)GO TO 9
      SENS(16)=DZ1/DW1
      SENS(17)=1.0
 24   DX1=0.
      DY1=0.
      DZ2=DZ1
      DZ1=SENS(16)*EW
      GO TO 8
C
C
 29   K=1
      MN=1
      L=2
      GO TO 50
```

```
C
 48    MN=1
C
C
C
C     FOR K EQUALS 1 THROUGH 4, SET UP INITIAL ESTIMATES OF
C     ELEMENTS OF SENSITIVITY MATRIX
C
C         SOLVE (X,Y'U,V) AND   (Z'W)   PROBLEMS SIMULTANEOUSLY BEFORE
C         SOLVING   (X,Y,Z'U,V,W)    PROBLEM
C
C
 50    IF(K.GE.4)GO TO 55
       IF(SENS(1).NE.0.)GO TO 56
       IF(K-2)51,52,53
 51    DX1=-.01*X
       DY1=0.
       DZ1=0.
       GO TO 8
C
 52    DX3=DX1
       DX1=0.
       DX2=0.
       DY1=-.01*Y
       GO TO 8
C
 53    DY2=DY1
       DY3=0.
       DY1=0.
       DZ1=-.01*Z
       GO TO 8
C
 55    DELVWA=DV2*DW3-DV3*DW2
       DELVWB=DV1*DW3-DV3*DW1
       DELVWC=DV1*DW2-DV2*DW1
       DELUWA=DU2*DW3-DW2*DU3
       DELUWB=DU1*DW3-DW1*DU3
       DELUWC=DU1*DW2-DW1*DU2
       DELUVA=DU2*DV3-DV2*DU3
       DELUVB=DU1*DV3-DV1*DU3
       DELUVC=DU1*DV2-DV1*DU2
       DENOM=DU1*DELVWA-DV1*DELUWA+DW1*DELUVA
       IF(DENOM.EQ.0.)GO TO 9
       SENS(2)=(DX1*DELVWA-DX2*DELVWB+DX3*DELVWC)/DENOM
       SENS(5)=(-DX1*DELUWA+DX2*DELUWB-DX3*DELUWC)/DENOM
       SENS(8)=(DX1*DELUVA-DX2*DELUVB+DX3*DELUVC)/DENOM
       SENS(3)=(DY1*DELVWA-DY2*DELVWB+DY3*DELVWC)/DENOM
       SENS(6)=(-DY1*DELUWA+DY2*DELUWB-DY3*DELUWC)/DENOM
       SENS(9)=(DY1*DELUVA-DY2*DELUVB+DY3*DELUVC)/DENOM
       SENS(4)=(DZ1*DELVWA-DZ2*DELVWB+DZ3*DELVWC)/DENOM
       SENS(7)=(-DZ1*DELUWA+DZ2*DELUWB-DZ3*DELUWC)/DENOM
       SENS(10)=(DZ1*DELUVA-DZ2*DELUVB+DZ3*DELUVC)/DENOM
       SENS(1)=1.
C    COMPUTE NEW X,Y,Z
 56    DX3=DX2
```

```
      DX2=DX1
      DY3=DY2
      DY2=DY1
      DZ3=DZ2
      DZ2=DZ1
C
      IF(L.GE.2) GO TO 70
C
 60   IF((M .EQ.1).OR.(ABS(EU).GT.QU ).OR.(ABS(EV).GT.QV ))GO TO 61
      M=1
      L=L+1
 61   DX1=SENS(2)*EU+SENS(5)*EV
      DY1=SENS(3)*EU+SENS(6)*EV
      IF((N .EQ.1).OR.(ABS(EW).GT.QW ))GO TO 62
      N=1
      L=L+1
C
 62   DZ1=SENS(10)*EW
C
      IF(L.LT.2) GO TO 8
C
 70   DX1=SENS(2)*EU+SENS(5)*EV+SENS(8)*EW
      DY1=SENS(3)*EU+SENS(6)*EV+SENS(9)*EW
      DZ1=SENS(4)*EU+SENS(7)*EV+SENS(10)*EW
C
 8    X=X-DX1
      Y=Y-DY1
      Z=Z-DZ1
      SU=U
      SV=V
      SW=W
      K=K+1
      CALL ROLLBK(I1)
      GO TO 10
C
C
    9 WRITE(6,102)I2
      CALL CHNXIT
      RETURN
C
 101  FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9/13X,'Y  =',E1
     .6.9,4X,'EV =',E16.9/13X,'Z  =',E16.9,4X,'EW =',E16.9)
 102  FORMAT(46H0**HUNT3 DID NOT CONVERGE TARGETING FOR EVENT ,A6,15HRUN
     1TERMINATED**)
 103  FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9,4X,'C11=',E16
     ..9/4X,'MN=',I2,4X,'Y  =',E16.9,4X,'EV =',E16.9/13X,'Z  =',E16.9,4X
     .,'EW =',E16.9)
 105  FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9,4X,'A11=',E16
     ..9,4X,'A12=',E16.9,4X,'A13=',E16.9/4X,'MN=',I2,4X,'Y  =',E16.9,4X,
     .'EV =',E16.9,4X,'A21=',E16.9,4X,'A22=',E16.9,4X,'A23=',E16.9/4X,'L
     . =',I2,4X,'Z  =',E16.9,4X,'EW =',E16.9,4X,'A31=',E16.9,4X,'A32=',E
     .16.9,4X,'A33=',E16.9)
      END
```

APPENDIX II

```
      SUBROUTINE HUNT4(X,Y,Z,R,U,V,W,Q,UD,VD,WD,QD,QU,QV,QW,QQ,I1,I2,
     . SENS)
C
      DIMENSION SENS(27)
C
C               SUBROUTINE HUNT4 FINDS VALUES OF X,Y,Z, AND R SUCH
C               THAT U,V,W, AND Q ARE EQUAL TO UD,VD,WD AND QD
C               WITHIN A TOLERANCE OF QU,QV,QW AND QQ, WHERE U,V,W
C               AND Q ARE DEPENDENT VARIABLES DEFINED AT EVENT I2
C               AND X,Y,Z AND R ARE CONTROL VARIABLES FOR THE FLIGHT
C               FROM I1 TO I2. IF THE SOLUTION DIVERGES, HUNT4
C               WILL STOP RUN AFTER 20 ITERATIONS.
C
C
      DATA ZERO/0.0/
C               SENS( 1)- FLAG TO INDICATE SENSITIVITY AVAILABLE
C               SENS( 2)- A11
C               SENS( 3)- A21
C               SENS( 4)- A31
C               SENS( 5)- A41
C               SENS( 6)- A12
C               SENS( 7)- A22
C               SENS( 8)- A32
C               SENS( 9)- A42
C               SENS(10)- A13
C               SENS(11)- A23
C               SENS(12)- A33
C               SENS(13)- A43
C               SENS(14)- A14
C               SENS(15)- A24
C               SENS(16)- A34
C               SENS(17)- A44
C
C               INITIALIZE
C
      K=1
      L=0
      M=0
      N=0
      MN=0
      EUMAX=1000000.
      EVMAX= 6.0
      EWMAX= 5.0
      EQMAX= 11.0
      IF(SENS(1).NE.0.0)GO TO 2
      DO 1 I=2,17
 1    SENS(I)=0.
 2    IF(SENS(22).NE.0.0)GO TO 4
      DO 3 I=18,21
 3    SENS(I)=0.
 4    IF(SENS(27).NE.0.0)GO TO 10
      DO 5 I=23,26
 5    SENS(I)=0.0
C               INTEGRATE TO EVENT I2 AND EVALUATE ERRORS
 10   CALL FLTINT(I1,I2)
```

```
      U=U+ZERO
      UD=UD+ZERO
      V=V+ZERO
      VD=VD+ZERO
      W=W+ZERO
      WD=WD+ZERO
      Q=Q+ZERO
      QD=QD+ZERO
      EU=U-UD
      EV=V-VD
      EW=W-WD
      EQ=Q-QD
      IF(MN) 15,16,16
15    IF(N.EQ.0) WRITE(6,103) K,X,EU,SENS(23),SENS(25),MN,Y,EV,SENS(24),
     .SENS(26),N,Z,EW
      IF(N.EQ.1) WRITE(6,104) K,X,EU,SENS(18),SENS(20),MN,Y,EV,SENS(19),
     .SENS(21),N,Z,EW
      GO TO 18
16     WRITE(6,101)K,X,EU,Y,EV,Z,EW,R,EQ
      IF(MN.EQ.0) GO TO 18
      WRITE(6,105) SENS(2),SENS(6),SENS(10),SENS(14),SENS(3),SENS(7),SEN
     .S(11),SENS(15),SENS(4),SENS(8),SENS(12),SENS(16),SENS(5),SENS(9),S
     .ENS(13),SENS(17)
18     IF((ABS(EU).LT.QU).AND.(ABS(EV).LT.QV).AND.(ABS(EW).LT.QW).AND.(A
     .BS(EQ).LT.QQ))RETURN
       IF(K.GE.30)GO TO 9
C
C                COMPUTE SENSITIVITIES
       DU4=DU3
       DU3=DU2
       DU2=DU1
       DU1=SU-U
       DV4=DV3
       DV3=DV2
       DV2=DV1
       DV1=SV-V
       DW4=DW3
       DW3=DW2
       DW2=DW1
       DW1=SW-W
       DQ4=DQ3
       DQ3=DQ2
       DQ2=DQ1
       DQ1=SQ-Q
      IF(MN) 20,19,50
19    IF((ABS(EU).LT.EUMAX).AND.(ABS(EV).LT.EVMAX).AND.(ABS(EW).LT.EWMAX
     .).AND.(ABS(EQ).LT.EQMAX)) GO TO 48
      MN=-1
C
C
C         SOLVE  (X,Y'U,V)  AND  (Z,R'W,Q)  PROBLEMS  SEPARATELY
C         BEFORE  SOLVING  (X,Y,Z,R'U,V,W,Q)   PROBLEM
C
C
C
```

```
20    IF(N.EQ.1) GO TO 30
      IF((ABS(EW).LT.QW    ).AND.(ABS(EQ).LT.QQ    )) GO TO 29
       IF(K.GT.2)GO TO 23
       IF(SENS(27).NE.0.0)GO TO 24
       IF(K.EQ.2)GO TO 22
21     DX1=0.0
       DY1=0.0
       DZ1=-.01*Z
       DR1=0.0
       GO TO 8
C
22     DZ2=DZ1
       DZ1=0.0
       DR1=-.01*R
       GO TO 8
C
23     DENOM3=DW1*DQ2-DW2*DQ1
       IF(DENOM3.EQ.0.0)GO TO 9
       SENS(23)=(DZ1*DQ2-DZ2*DQ1)/DENOM3
       SENS(25)=(DW1*DZ2-DW2*DZ1)/DENOM3
       SENS(24)=(DQ2*DR1-DQ1*DR2)/DENOM3
       SENS(26)=(DW1*DR2-DW2*DR1)/DENOM3
       SENS(27)=1.
24     DX1=0.
       DY1=0.
       DZ2=DZ1
       DR2=DR1
       DZ1=SENS(23)*EW+SENS(25)*EQ
       DR1=SENS(24)*EW+SENS(26)*EQ
       GO TO 8
C
29     K=1
       N=1
30    IF((ABS(FU).LT.QU    ).AND.(ABS(FV).LT.QV    )) GO TO 49
       IF(K.GT.2)GO TO 33
       IF(SENS(22).NE.0.0)GO TO 34
       IF(K.EQ.2)GO TO 32
31     DX1=-.01*X
       DY1=0.
       DZ1=0.
       DR1=0.
       GO TO 8
C
32     DX2=DX1
       DX1=0.
       DY1=-.01*Y
       GO TO 8
C
33     DENOM2=DU1*DV2-DU2*DV1
       IF(DENOM2.EQ.0.0)GO TO 9
       SENS(18)=(DX1*DV2-DX2*DV1)/DENOM2
       SENS(20)=(DU1*DX2-DU2*DX1)/DENOM2
       SENS(19)=(DV2*DY1-DV1*DY2)/DENOM2
       SENS(21)=(DU1*DY2-DU2*DY1)/DENOM2
       SENS(22)=1.0
```

```
 34     DZ1=0.0
        DR1=0.0
        DX2=DX1
        DY2=DY1
        DX1=SENS(18)*EU+SENS(20)*EV
        DY1=SENS(19)*EU+SENS(21)*EV
        GO TO 8
C
C
C
C        SOLVE  (X,Y'U,V)  AND  (Z,R'W,Q)  PROBLEMS  SIMULTANEOUSLY
C        BEFORE  SOLVING  (X,Y,Z,R'U,V,W,Q)  PROBLEM
C
C
 48     MN=1
        GO TO 50
 49      K=1
        MN=1
        L=2
C
C
 50     IF(K.GE.5) GO TO 55
         IF(SENS(1).NE.0.0)GO TO 56
         IF(K.EQ.4)GO TO 54
         IF(K-2)51,52,53
 51      DX1=-.01*X
         DY1=.0
         DZ1=.0
         DR1=.0
         GO TO 8
C
 52      DX4=DX1
         DX1=0.
         DX2=0.
         DX3=0.
         DY1=-.01*Y
         GO TO 8
C
 53      DY3=DY1
         DY1=0.
         DY2=0.
         DY4=0.
         DZ1=-.01*Z
         GO TO 8
C
C
 54      DZ2=DZ1
         DZ1=0.
         DZ3=0.
         DZ4=0.
         DR1=-.01*R
         DR2=0.
         DR3=0.
         DR4=0.
         GO TO 8
```

```
C
55    DELUVC=DU3*DV4-DU4*DV3
      DELUVA=DU2*DV3-DV2*DU3
      DELUVB=DU2*DV4-DU4*DV2
      DELQVA=DV2*DQ3-DQ2*DV3
      DELQVB=DV2*DQ4-DQ2*DV4
      DELQVC=DV3*DQ4-DQ3*DV4
      DELVWA=DV2*DW3-DV3*DW2
      DELVWB=DV2*DW4-DW2*DV4
      DELVWC=DV3*DW4-DW3*DV4
      DELQWA=DW2*DQ3-DQ2*DW3
      DELQWB=DW2*DQ4-DQ2*DW4
      DELQWC=DW3*DQ4-DQ3*DW4
      DELQUA=DU2*DQ3-DQ2*DU3
      DELQUB=DU2*DQ4-DQ2*DU4
      DELQUC=DU3*DQ4-DQ3*DU4
      DELUWA=DU2*DW3-DW2*DU3
      DELUWB=DU2*DW4-DW2*DU4
      DELQUA=DQ3*DU2-DQ2*DU3
      DELQUB=DU2*DQ4-DQ2*DU4
      DELUWC=DU3*DW4-DW3*DU4
      ADJ11=DV2*DELQWC-DW2*DELQVC+DQ2*DELVWC
      ADJ12=DV1*DELQWC-DW1*DELQVC+DQ1*DELVWC
      ADJ13=DV1*DELQWB-DW1*DELQVB+DQ1*DELVWB
      ADJ14=DV1*DELQWA-DW1*DELQVA+DQ1*DELVWA
      ADJ21=DU2*DELQWC-DW2*DELQUC+DQ2*DELUWC
      ADJ22=DU1*DELQWC-DW1*DELQUC+DQ1*DELUWC
      ADJ23=DU1*DELQWB-DW1*DELQUB+DQ1*DELUWB
      ADJ24=DU1*DELQWA-DW1*DELQUA+DQ1*DELUWA
      ADJ31=DU2*DELQVC-DV2*DELQUC+DQ2*DELUVC
      ADJ32=DU1*DELQVC-DV1*DELQUC+DQ1*DELUVC
      ADJ33=DU1*DELQVB-DV1*DELQUB+DQ1*DELUVB
      ADJ34=DU1*DELQVA-DV1*DELQUA+DQ1*DELUVA
      ADJ41=DU2*DELVWC-DV2*DELUWC+DW2*DELUVC
      ADJ42=DU1*DELVWC-DV1*DELUWC+DW1*DELUVC
      ADJ43=DU1*DELVWB-DV1*DELUWB+DW1*DELUVB
      ADJ44=DU1*DELVWA-DV1*DELUWA+DW1*DELUVA
      DENOM=DU1*ADJ11-DV1*ADJ21+DW1*ADJ31-DQ1*ADJ41
      IF(DENOM.EQ.0.)GOTO9
      SENS(2)=(DX1*ADJ11-DX2*ADJ12+DX3*ADJ13-DX4*ADJ14)/DENOM
      SENS(3)=(DY1*ADJ11-DY2*ADJ12+DY3*ADJ13-DY4*ADJ14)/DENOM
      SENS(4)=(DZ1*ADJ11-DZ2*ADJ12+DZ3*ADJ13-DZ4*ADJ14)/DENOM
      SENS(5)=(DR1*ADJ11-DR2*ADJ12+DR3*ADJ13-DR4*ADJ14)/DENOM
      SENS(6)=(-DX1*ADJ21+DX2*ADJ22-DX3*ADJ23+DX4*ADJ24)/DENOM
      SENS(7)=(-DY1*ADJ21+DY2*ADJ22-DY3*ADJ23+DY4*ADJ24)/DENOM
      SENS(8)=(-DZ1*ADJ21+DZ2*ADJ22-DZ3*ADJ23+DZ4*ADJ24)/DENOM
      SENS(9)=(-DR1*ADJ21+DR2*ADJ22-DR3*ADJ23+DR4*ADJ24)/DENOM
      SENS(10)=(DX1*ADJ31-DX2*ADJ32+DX3*ADJ33-DX4*ADJ34)/DENOM
      SENS(11)=(DY1*ADJ31-DY2*ADJ32+DY3*ADJ33-DY4*ADJ34)/DENOM
      SENS(12)=(DZ1*ADJ31-DZ2*ADJ32+DZ3*ADJ33-DZ4*ADJ34)/DENOM
      SENS(13)=(DR1*ADJ31-DR2*ADJ32+DR3*ADJ33-DR4*ADJ34)/DENOM
      SENS(14)=(-DX1*ADJ41+DX2*ADJ42-DX3*ADJ43+DX4*ADJ44)/DENOM
      SENS(15)=(-DY1*ADJ41+DY2*ADJ42-DY3*ADJ43+DY4*ADJ44)/DENOM
      SENS(16)=(-DZ1*ADJ41+DZ2*ADJ42-DZ3*ADJ43+DZ4*ADJ44)/DENOM
      SENS(17)=(-DR1*ADJ41+DR2*ADJ42-DR3*ADJ43+DR4*ADJ44)/DENOM
```

```
      SENS(1)=1.
C     COMPUTE NEW X AND Y
 56   DX4=DX3
      DX3=DX2
      DX2=DX1
      DY4=DY3
      DY3=DY2
      DY2=DY1
      DZ4=DZ3
      DZ3=DZ2
      DZ2=DZ1
      DR4=DR3
      DR3=DR2
      DR2=DR1
      IF(L.GE.2)GO TO 70
 60   IF((M .EQ.1).OR.(ABS(EU).GT.QU ).OR.(ABS(EV).GT.QV )) GO TO 61
      M=1
      L=L+1
C
 61   DX1=SENS(2)*EU+SENS(6)*EV
      DY1=SENS(3)*EU+SENS(7)*EV
C
      IF((N .EQ.1).OR.(ABS(EW).GT.QW ).OR.(ABS(EQ).GT.QQ )) GO TO 62
      N=1
      L=L+1
C
 62   DZ1=SENS(12)*EW+SENS(16)*EQ
      DR1=SENS(13)*EW+SENS(17)*EQ
C
      IF(L.LT.2)GO TO 8
C
 70   DX1=SENS(2)*EU+SENS(6)*EV+SENS(10)*EW+SENS(14)*EQ
      DY1=SENS(3)*EU+SENS(7)*EV+SENS(11)*EW+SENS(15)*EQ
      DZ1=SENS(4)*EU+SENS(8)*EV+SENS(12)*EW+SENS(16)*EQ
      DR1=SENS(5)*EU+SENS(9)*EV+SENS(13)*EW+SENS(17)*EQ
 8    X=X-DX1
      Y=Y-DY1
      Z=Z-DZ1
      R=R-DR1
      SU=U
      SV=V
      SW=W
      SQ=Q
      K=K+1
      CALL ROLLBK(I1)
      GO TO 10
C
 9    WRITE(6,102)I2
      CALL CHNXIT
      RETURN
C
 101  FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9/13X,'Y  =',E1
     .6.9,4X,'EV =',E16.9/13X,'Z  =',E16.9,4X,'EW =',E16.9/13X,'R  =',E1
     .6.9,4X,'EQ =',E16.9)
 102  FORMAT(46H0**HUNT4 DID NOT CONVERGE TARGETING FOR EVENT, A6,19H -
```

```
     . RUN TERMINATED**)
103   FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9,4X,'C11=',F16
     ..9,4X,'C12=',F16.9/4X,'MN=',I2,4X,'Y  =',E16.9,4X,'EV =',F16.9,4X,
     .'C21=',F16.9,4X,'C22=',F16.9/4X,'N =',I2,4X,'Z  =',E16.9,4X,'EW ='
     .,F16.9)
104   FORMAT(//4X,'K =',I2,4X,'X  =',E16.9,4X,'EU =',E16.9,4X,'B11=',F16
     ..9,4X,'B12=',F16.9/4X,'MN=',I2,4X,'Y  =',E16.9,4X,'EV =',F16.9,4X,
     .'B21=',F16.9,4X,'B22=',F16.9/4X,'N =',I2,4X,'Z  =',E16.9,4X,'EW ='
     .,F16.9)
105   FORMAT(//13X,'A11=',F16.9,4X,'A12=',F16.9,4X,'A13=',F16.9,4X,'A14=
     .',F16.9/13X,'A21=',F16.9,4X,'A22=',F16.9,4X,'A23=',F16.9,4X,'A24='
     .,F16.9/13X,'A31=',F16.9,4X,'A32=',F16.9,4X,'A33=',F16.9,4X,'A34=',
     .F16.9/13X,'A41=',F16.9,4X,'A42=',F16.9,4X,'A43=',F16.9,4X,'A44=',F
     .16.9)
      END
```

BELLCOMM, INC.

Subject: Subroutines HUNT3 and HUNT4 for Trajectory Targeting with BCMASP - Case 610

From: P. H. Whipple

DISTRIBUTION LIST

NASA Headquarters

Messrs. H. Cohen/MLR
P. E. Culbertson/MLA
J. H. Disher/MLD
L. K. Fero/MLV
J. P. Field/MLP
E. L. Harkleroad/MLO
T. A. Keegan/MA-2
M. Savage/MLT
W. C. Schneider/ML

Bellcomm, Inc.

Messrs. A. P. Boysen, Jr.
D. A. Chisholm
D. R. Hagner
B. T. Howard
J. Z. Menard
I. M. Ross
J. W. Timko
R. L. Wagner

Division 101 Supervision
All Members Departments 1021, 1022, 1024, 1025
Department 1024 File
Central Files
Library